What is Claimed:

1. A method of generating a manifest that governs the execution of a software object, the method comprising:

receiving a specification indicative of requirements for the execution of the software object, the specification referring to one or more components;

generating a manifest based on said specification, including accessing said one or more components, said manifest comprising one or more rules governing what may be loaded into an address space of the software object.

2. The method of claim 1, wherein said specification identifies one or more modules, and wherein generating the manifest comprises including, in said manifest, the identities of the one or more modules identified in the specification.

3. The method of claim 2, wherein said specification indicates that a first one of said one or more modules may be loaded into the address space of the software object, and wherein generating the manifest comprises including the identity of said first one of said one or more modules on list of acceptable modules.

4. The method of claim 2, wherein said specification indicates that a first one of said one or more modules may not be loaded into the address space of the software object, and wherein generating the manifest comprises including in the manifest a datum that identifies said first one of said one or more modules.

5. The method of claim 2, wherein said datum comprises a hash of said first one of said one or more modules.

6. The method of claim 2, wherein said specification indicates whether said manifest will contain hashes of said one or more modules.

7. The method of claim 1, wherein said one or more components comprise a key, wherein said specification indicates either that modules signed with said key may be loaded into said address space or that modules signed with said key may not be loaded into said address space, and wherein generating said manifest comprises:

retrieving said key from a file identified in said specification; and

including said key in said manifest.

8. The method of claim 1, wherein said one or more components comprise a module, wherein said specification indicates that said module may not be loaded into said address space, and wherein generating said manifest comprises:

computing a hash of said module; and

including said hash in said manifest.

9. The method of claim 1, wherein said generating act comprises:

based on said specification, creating a data structure representative of said specification; and

generating said manifest based on said data structure.

10. The method of claim 1, further comprising:

receiving a key associated with a vendor or distributor of said software object;

signing said manifest with said to produce a digital signature; and

including said digital signature in said manifest.

11. The method of claim 1, further comprising:

using a hardware security module to sign said manifest, said hardware security module being adapted to apply a key associated with a vendor or distributor of said software object without revealing said key outside said hardware security module.

12. A computer-readable medium encoded with computer-executable instructions to perform a method of generating a manifest, the method comprising:

parsing a specification of requirements to be included in the manifest, the requirements defining a policy that governs what can be loaded into an address space of a software object associated with the manifest;

accessing one or more components that are identified by the specification and that are external to the specification; and

generating a manifest based on at least one of the accessed objects.

13. The computer-readable medium of claim 12, wherein said one or more components comprise an executable module, and wherein generating said manifest comprises:

including in said manifest an identification of said executable module and an indication that either:

said executable module may be loaded into said address space; or

said executable module may not be loaded into said address space.

14. The computer-readable medium of claim 13, wherein said identification of said executable module comprises a hash of said executable module.

15. The computer-readable medium of claim 12, wherein said one or more components comprise a key, wherein said specification indicates either that modules signed with said key may be loaded into said address space or that modules signed with said key may not be loaded into said address space, and wherein generating said manifest comprises:

retrieving said key from a file identified in said specification; and

including said key in said manifest.

16. The computer-readable medium of claim 12, wherein the method further comprises:

receiving a key associated with a vendor or distributor of said software object;

signing said manifest with said to produce a digital signature; and

including said digital signature in said manifest.

17. A method of specifying constraints on the use of software comprising:

creating a specification concerning what may be loaded into an address space of the software, the specification referring to one or more components that are external to the software and external to the specification;

using a manifest generation tool to generate a manifest based on the specification, wherein the manifest generation tool does at least one of:

including, in said manifest, data from one of said one or more components; or

computing a value based on one of said one or more components and including the computed value in said manifest; and

distributing the generated manifest with the software, wherein the manifest comprises rules describing what may be loaded into the address space of the software.

18. The method of claim 17, wherein said one or more components comprises a module, wherein said specification indicates either that said module may be loaded into said address space or that said module may not be loaded into said address space, and wherein said manifest generation tool does at least one of:

including an identifier of said module in said manifest; or

computing a hash of said module and including the hash in said manifest.

19. The method of claim 17, wherein said one or more components comprise a key, wherein said specification indicates either that modules signed with said key may be loaded into said address space or that modules signed with said key may not be loaded into said address space, and wherein said manifest generation tool retrieves said key from a file identified in said specification, and includes a certificate for said key in said manifest.

20. The method of claim 17, wherein said manifest generation tool creates an intermediate data structure representative of said specification, and generates said manifest based on said intermediate data structure.

21. The method of claim 17, wherein the method further comprises:

receiving a key from further comprising:

receiving a key associated with a vendor or distributor of the software;

signing said manifest with said to produce a digital signature; and

including said digital signature in said manifest.

22. The method of claim 17, further comprising:

using a hardware security module to sign said manifest, said hardware security module being adapted to apply a key associated with a vendor or distributor of the software without revealing said key outside said hardware security module.

23. A system for generating a manifest comprising:

a first parser that receives a manifest specification indicative of requirements for a manifest, the first parser generating a representation of said requirements, said requirements relating to what may be loaded into an address space of a software object, said specification referring to one or more components external to said software and external to said specification;

a first manifest generator that generates a manifest based on said representation and includes in said manifest information contained in, or computed based on, said one or more components.

24. The system of claim 23, wherein said one or more components comprise a module, and wherein said first manifest generator generates said manifest by including, in said manifest, a datum that identifies said module.

25. The system of claim 23, wherein said datum comprises a hash of said module.

26. The system of claim 23, wherein said one or more components comprise a key, wherein said specification indicates either that modules signed with said key may be loaded into said address space or that modules signed with said key may not be loaded into said address space, and wherein

said first manifest generator retrieves said key from a file identified in said specification and includes said key in said manifest.

27. The system of claim 23, wherein said first manifest generator generates a digital signature for said manifest by signing said manifest with a key associated with a vendor or distributor of said software object, and includes said digital signature in said manifest.

28. The system of claim 27, further comprising:

a hardware security module that applies said key without revealing said key outside said hardware security module, said first manifest generating using said hardware security module to generate said digital signature.

29. The system of claim 23, further comprising:

a second parser that receives a manifest specification indicative of requirements for a manifest, the second parser generating a representation of said requirements in the same format as said first parser,

wherein said first parser parses specifications in a first format and second parser parses specifications in a second format different from said first format, and wherein first manifest generator generates said manifest based on a representation produced either by said first parser or said second parser.

30. The system of claim 23, further comprising:

a second manifest generator that generates a manifest based on said representation, wherein said first manifest generator generates a manifest in a first format and second manifest generator generates a manifest in a second format different from said first format.